ログファイルを move したらえらいこっちゃ！？

運用監視業務をしているとログファイルのローテーションなんてことを
する機会が多いと思います。

ログファイルのローテーションにも色々と方法が考えられますが今回は
一般的な move でファイル名を変名しするパターンの際の注意事項です。

さて、定期的に出力されているログファイル (syslog など) を UNIX の mv(move) コマンドで
ファイル名を変名したらどういうことが起こるのでしょうか？

この事を知っていないとログ監視する際にえらいことになってしまいますので
知っていると良いと思います。

ログファイルの move 検証

では、Linux の messages(syslog) ファイルを move したらどうなるのか検証してみます。

■１．現在の状況確認とログファイルの move

ここではファイルの inode 番号が重要なポイントですのでコマンドは inode 番号を
表示する為に「-i」オプションをつけて実行しています。
※一番左側の数字が inode 番号になります。

```
[root@asiapacific log]# ls -li messages*
2785680 -rw-------  1 root root  29144  7月 21 23:02 messages
2785622 -rw-------  1 root root 154460  7月 13 01:38 messages.1
2785682 -rw-------  1 root root  60483  5月  1 11:01 messages.2
2785562 -rw-------  1 root root  27860  4月 28 18:55 messages.3
2785331 -rw-------  1 root root 206133  4月 13 19:40 messages.4
[root@asiapacific log]#
[root@asiapacific log]# mv messages messages.old
```

inode 番号『2785680』の messages ファイルを move してみました。
さて inode 番号はどうなっているのでしょうか？

■２．move 後のファイル確認 (inode 番号)

move したろところ inode 番号は変わらずにファイル名だけ変名されています！！

```
[root@asiapacific log]# ls -li messages*
2785622 -rw-------  1 root root 154460  7月 13 01:38 messages.1
2785682 -rw-------  1 root root  60483  5月  1 11:01 messages.2
2785562 -rw-------  1 root root  27860  4月 28 18:55 messages.3
2785331 -rw-------  1 root root 206133  4月 13 19:40 messages.4
2785680 -rw-------  1 root root  29144  7月 21 23:02 messages.old
[root@asiapacific log]#
```

■３．messages ファイルに文字列を書き込んでみます。

logger コマンドで messages に文字列を書き込むテストをしてみます。
logger が何するのかよく分からない人は『man logger』などで
調べてみてください。運用監視系ではよく使うコマンドの一つだと思いますので。

```
[root@asiapacific log]# logger "syslog moveing `date`"
[root@asiapacific log]# tail messages.old
```

```
Jul 21 23:00:40 asiapacific kernel: fb0: VGA16 VGA frame buffer device
Jul 21 23:00:41 asiapacific fstab-sync[3702]: removed all generated mount points
Jul 21 23:00:42 asiapacific fstab-sync[3727]: added mount point /media/cdrecorder  for /dev/hdb
Jul 21 23:04:38 asiapacific root: syslog moveing 2007年  7月 21日 土曜日 23:04:38 JST
[root@asiapacific log]#
```

なんと変名したはずの『messages.old』ファイルに書き込まれているではありませんかっ！
以下の部分がその部分になります。

```
Jul 21 23:04:38 asiapacific root: syslog moveing 2007年  7月 21日 土曜日 23:04:38 JST
```

■４．syslog デーモンをリフレッシュして新たに messages ファイルを作成する。

元の messages ファイルは move で messages.old に変名されてしまったのでファイルがありません。

「３.」の事象から今後は messages.old の方へログが出力されることになります。
もし、messages ファイルをログ監視していたら大変っ！監視されていないことになります！
ってことで、syslog デーモンを『kill -HUP』コマンドでリフレッシュします。
『kill -HUP』の代わりに Linux なら『service syslod restart』などでも構わないと思います。
※『service syslod restart』すると PID が変わるので PID によるプロセス監視などを実装しているとこれまた影響ありありなわけです。。。 >_<

```
[root@asiapacific log]# ps -ef | grep syslog | grep -v grep
root      2690     1  0 23:00 ?        00:00:00 syslogd -m 0
canna     3046     1  0 23:00 ?        00:00:00 /usr/sbin/cannaserver -syslog -u   canna
[root@asiapacific log]# kill -HUP 2690
[root@asiapacific log]# ps -ef | grep syslog | grep -v grep
root      2690     1  0 23:00 ?        00:00:00 syslogd -m 0
canna     3046     1  0 23:00 ?        00:00:00 /usr/sbin/cannaserver -syslog -u canna
[root@asiapacific log]#
```

『kill -HUP』ならプロセス ID(PID) が変更されていないですね！

■５．syslog デーモンのリフレッシュ

syslog デーモンプロセスのリフレッシュ後にもう一度『logger』コマンドで
文字列を書き込んでみよう。

```
[root@asiapacific log]# ll messages*
-rw-------  1 root root     49  7月 21 23:05 messages
-rw-------  1 root root  29233  7月 21 23:04 messages.old
```

リフレッシュしたら新しく messages が作成されていますね。

```
[root@asiapacific log]# logger "syslog moveing `date`"
[root@asiapacific log]# cat messages
Jul 21 23:05:07 asiapacific syslogd 1.4.1: restart.
Jul 21 23:05:23 asiapacific root: syslog moveing 2007年  7月 21日 土曜日 23:05:23 JST
[root@asiapacific log]#
```

今度は新しく作成された messages ファイルの方にキチンと書き込まれていますね。
一安心です。

■６．この検証で分かったポイントは？

・mv でファイルを変名しても inode 番号は変わらない。

・syslod デーモンは inode をポインタしているようでファイルを変名しても旧ファイル名の inode 番号を持つファイルにログを書き込む。
・デーモンをリフレッシュする際に kill -HUP を使うことで PID を変えずにリフレッシュ出来る。

日々業務で忙しいですが、時間があれば inode 番号、ファイルサイズなどをチェックし、
オリジナルのログファイルを切り出すスクリプトやログローテーションのスクリプトなども紹介していきたいと思います。